NERSC-6 Benchmark Instructions and Result Tables

Introduction

Benchmarks play a critical role in evaluation of the offered system. The NERSC-6 benchmarks serve three purposes:

1. The benchmarks have been carefully chosen to represent characteristics of the expected NERSC-6 workload, which consists of solving complex scientific problems using diverse computational techniques at high degrees of parallelism.

2. The benchmarks give the Offeror the opportunity to provide the University concrete data associated with the performance and scalability of the proposed system on applications that NERSC and the DOE consider programmatically important.

3. The benchmarks will be used as an integral part of the system acceptance test and as a measurement of performance throughout the operational lifetime of the system.

If selected, the Offeror shall be required to meet benchmark performance levels reported in their RFP response as a condition of acceptance and throughout the life of the subcontract. The Offeror may be required to demonstrate other performance metrics not mentioned herein as part of a negotiated statement of work.

The NERSC-6 benchmarks comprise tests at varying levels of the benchmark hierarchy that range from system component-level tests and kernels to Full Application Benchmarks. Results from the Full Application Benchmarks are used to derive the Sustained System Performance (SSP) metric, which is used to assess the useful potential of the offered system for NERSC's anticipated scientific application workload. The Offeror should pay particular attention to the SSP calculation, as it is one of the key metrics for system evaluation in the NERSC-6 procurement.

Observed benchmark performance shall be obtained from a system configured as closely as possible to the proposed system. Since NERSC-6 will support highly parallel computation, it is critical that the Offeror provide observed application performance using the extra-large and large test inputs as well as the medium for the Full Application Benchmarks. The largest jobs in the benchmark suite should not be interpreted as the limit for the job concurrency for the target system. The target system will support jobs that range in scale from the supplied SSP benchmark scale all the way up to jobs that span the entire system. Performance projections are permissible if they are derived from a similar system that is considered an earlier generation system. Projections shall be rigorously derived, thoroughly documented and easily understood. In the tables below,

the "Proposed" column refers to the value for the full, proposed system, whether benchmarked or projected. The University will be the sole judge of the validity of any projected results. For solutions that use multiple phases of technology to offer the best value, the Offeror should duplicate the Result Tables in this document and provide information for each phase.

Relation to DOD HPCMP TI-09

One full application benchmark, GAMESS, uses the same input files and source code as the DOD HPCMP TI-09. The NERSC "medium" size corresponds to the DOD "standard" input and the NERSC "large" size corresponds to the DOD "large" input.

Submission Guidelines

Most benchmark results (or projections including original results) for the proposed system shall be recorded in the tables provided at the end of this document. Some results are to be recorded in tables provided in the benchmark tar files. A paper version of all completed tables shall be submitted as part of the RFP response. Additionally, the Offeror shall submit electronically all completed tables, benchmark codes and output files and documentation on any code optimizations or configuration changes on a CD or similar medium. The submitted source shall be in a form that can be readily compiled on the proposed system. Do not include object and executable files, core dump files or large binary data files in the electronic submission. An audit trail showing any changes made to the benchmark codes must be supplied and it must be sufficient for the University to determine that the changes made conform to the spirit of the benchmark and do not violate any specific restrictions on the various benchmark codes.

If performance projections are used, this must be clearly indicated. The output files on which the projections are based, and a description of the projection method must be included. In addition, each system used for benchmark projections must be described in Table 3 below. Each projection in benchmarks results tables must indicate on which system the benchmark was originally run. Enter the corresponding letter of the "System" column of Table 3 into the "System" column of the benchmark result tables. For solutions that use multiple phases of technology to offer the University the best value, the Offeror should duplicate the tables and provide the performance information for each phase.

Consistency

As described in the NERSC-6 RFP Letter, NERSC expects consistent and reproducible execution times in multi-user production mode for the proposed system. The Offeror shall document the amount of run time variation that the system shall have for production mode by including the expected coefficient of variation in the elapsed time in the tables. The coefficient of variation is defined as the standard deviation of run times divided by the mean of the same run times for a minimum of five consecutive runs. In addition, the Offeror shall document the coefficient of variation for the SSP in Table 8, for the

benchmarks comprising the SSP running in production mode. Production mode refers to the multi-user, general use environment at NERSC.

Run Rules

Additional run rules may be included with the individual benchmark source code distribution, supplying specific requirements and instructions for compiling, executing, verifying numerical correctness and reporting results for each benchmark. Benchmark performance shall only be accepted from runs that exhibit correct execution. Only software tools and libraries that will be included for general use in the proposed system as supported product offerings are permissible for building and executing the benchmarks.

Message passing programs must be built using an implementation that supports 64-bit virtual memory pointers and a thread-safe communication library that implements the MPI standard.

Benchmark Descriptions

Lower Level Tests

The Lower Level Tests, listed in Table 1, are simple, focused tests that are easily compiled and executed. The results allow a uniform comparison of features and provide an estimation of system balance. Descriptions and requirements for each test are included in the source distribution. The results for the proposed system shall be recorded in Table 4, except for IOR, Metabench, and NetPerf the results for which are recorded in tables provided with the source distribution. In the event that benchmark results are being projected, columns "Benchmarked" and "Proposed" should be filled out. For benchmarks where the results are only required on the CD submission, the "System" column must still be filled out in Table 4, and both sets (run and projected) of results included on the CD. For these runs an entry should be made in Table 4 stating the benchmark was executed.

Modifications to the Lower Level Tests are only permissible to enable correct execution on the target platform. No changes related to optimization are permissible except in the case of the NAS FT benchmark where the values for `fftblock_default` and `fftblockpad_default` may be changed to suit the target architecture.

Table 1. Lower Level Tests

Benchmark	Purpose
NAS Parallel 2.3, serial-packed	Assess Sequential Performance
NAS Parallel 2.4 Class D, 256 tasks	Parallel Performance/Interconnect
NAS Parallel UPC Class D	PGAS Performance/One-sided-Messages
STREAM	Memory Bandwidth
AMR Elliptic	Memory/interconnect perf + OS Jitter
PSNAP	OS Jitter
Multipong	Interconnect performance
Netperf	IP interface performance
IOR	Sequential & Parallel I/O performance
Metabench	Filesystem Metadata Server Performance

All tests are to be run in fully packed mode unless otherwise described below. In architectures with multiple cores per node, "fully packed" means that the number of instances or MPI tasks per node shall at least equal the total number of cores available on the node.

All NPB v2.3 serial benchmarks must be run so that multiple instances execute simultaneously on the node. The number of instances shall equal the number of physical cores on the node.

The NPB UPC FT Class D benchmark must execute with 256 UPC threads. The Offeror may choose the number of nodes.

The PSNAP benchmark must execute on all available core on the benchmark system. The operating system used for the PSNAP run(s) must be configured as the system would be delivered to and used at NERSC for regular, production purposes.

Special rules regarding packing apply to the STREAM and MULTIPONG benchmarks; see the individual README files for details.

For the MULTIPONG benchmark the terms "Minimum Latency" and "Maximum Latency" refer to the best case and worse case, respectively for the Offeror's interconnect topology.

Full Application Tests

The Full Application Benchmarks are a representation of the NERSC workload and span a variety of algorithmic and computational characteristics. The list of application benchmarks is shown in Table 2. Documentation for each application is included with the source distribution. For most applications there are two problem sizes provided for each application, a medium size and a large size as shown in Table 5. The exception is MILC, for which an extra-large size is required. The purpose of this extra-large size is to make the SSP metric more representative of the actual NERSC workload.

Table 2: Full Application Benchmarks

Application	Discipline
CAM	Climate
GAMESS	Quantum Chemistry
GTC	Plasma Physics
IMPACT-T	Particle Accelerator
MAESTRO	Astrophysics
MILC	Lattice QCD
PARATEC	Material Science

Two cases for running the seven NERSC Full Application benchmarks are described below, a 'Base Case,' which will be an important basis for University comparison amongst proposed systems, and an optional 'Optimized Performance' case that allows the Offeror broader latitude to optimize code and demonstrate the best-case performance potential of the system. ***It is extremely important for the Offeror to provide results for each benchmark for at least the base case!***

Base Case

The base case limits the scope of optimization and the allowable concurrency to prescribed values. Certain minimal exceptions are allowed for hardware multithreading and if there is insufficient memory per node to execute the application. The base case also limits the parallel programming model to MPI only. Each of these points is covered in more detail below. In the Base Case for all Full Application runs, modifications are permissible only to enable porting and correct execution on the target platform. No changes related to optimization are permissible. Library routines may be used as long as they currently exist in an Offeror's supported set of general or scientific libraries, and must be in such a set when the system is delivered. As well, the libraries must not specialize or limit the applicability of the benchmark nor violate the measurement goals of the particular benchmark. Source preprocessors, execution profile feedback optimizers, etc. are allowed as long as they are, or will be, available and supported as part of the compilation system for the full-scale systems. Only publicly available and documented

compiler switches shall be used. Compiler optimizations will be allowed only if they do not increase the runtime or artificially increase the delivered FLOP/s rate by performing non-useful work.

For each benchmark code a target processor count is given for two problem sizes (three for MILC) and the Offeror should submit results at the target concurrencies if it is possible to fit the benchmark on its target number of processors.

If a benchmark will not run on its target number of processors due to memory limitations, the Offeror may use the least number of additional processors necessary. The Offeror must still solve the same global problem, using the same input files as for the target concurrency when the MPI concurrency is higher than the original target. For two codes (CAM and IMPACT-T) the number of processors to be used is specified in the input files and so if a larger number of processors than the target is required the input files may be modified accordingly; other than that, no changes to the input files are allowed.

For all Base Case runs the benchmarks must be executed in a *fully-packed* manner on the computational nodes. In this mode all the Full Applications will run as single-threaded MPI-only applications. In architectures with multiple cores per node, the number of MPI tasks per node shall at least equal the total number of cores available on the node. GAMESS is an exception because it implements its own communication layer; see the README.

It is permissible for applications to run with more than one MPI task per core if the proposed system has the hardware capability to run multiple tasks, and the capability can be activated with a simple environment setting that would be available to NERSC users. To use hardware multithreading, the Offeror must first start with the NERSC target concurrency given in the tables and then expand MPI concurrency to occupy hardware threads. For example, for 2-way hardware multithreading, the Offeror must first start with the target concurrency (512 for the medium MAESTRO run, for example) and then expand to 1024 in order to engage the 2-way hardware threading.

Base case results should be entered in **Table 5A**, below.

Optional Optimized Case

An optional optimized case has been added to allow the Offeror to highlight the features and benefits of the proposed system by submitting benchmarking results obtained through a variety of optimizations. This case applies only to the seven Full Application Benchmarks and it applies to all sizes (subject to the constraints below). The Offeror may choose to optimize the source code for data layout and alignment or to enable specific hardware or software features that may include (but are not limited to):

- Using Hybrid OpenMP+MPI for concurrency;
- Using vendor-specific hardware features to accelerate code;
- Running the benchmarks at a higher or lower concurrency than the targets;

- Running at the same concurrency as the targets but in an "unpacked" mode;
- Any combination of the above.

Note: ***When running in an unpacked mode, the number of tasks used in the SSP calculation for that application must be calculated using the total number of processors blocked from other use***. For example, if the scheduling unit is a node, all the cores in all the nodes assigned to the job must be counted as being used. The Offeror should determine if the SSP increases or decreases when running in an unpacked mode before submitting results.

Wholesale changes to the parallel algorithms are also permitted as long as the full capabilities of the code are maintained; the code can still pass validation tests; and the underlying purpose of the benchmark is not compromised. As many changes to the code may be made as wanted so long as the following conditions are met:

- All simulation parameters such as grid size, number of particles, etc., must not be changed.
- The optimized code execution must still result in correct numerical results.
- Any code optimizations must be available to the general NERSC user community, either through a system library or a well-documented explanation of code improvements.
- Any library routines used must currently exist in an Offeror's supported set of general or scientific libraries, or must be in such a set when the system is delivered, and must not specialize or limit the applicability of the benchmark nor violate the measurement goals of the particular benchmark.
- Source preprocessors, execution profile feedback optimizers, etc. are allowed as long as they are, or will be, available and supported as part of the compilation system for the full-scale systems.
- Only publicly available and documented compiler switches shall be used.
- Finally, the same code optimizations must be made for all runs of a benchmark. For example, one set of code optimizations may not be made for the smaller concurrency while a different set of optimizations are made for the larger concurrency.

Any specific code changes and the runtime configuration used must be clearly documented with a complete audit trail and all supporting documentation included in the CD submission. The University will be the final judge of whether optimizations will be acceptable. Optional Optimized case results should be entered in **Table 5B**, below.

Application-Level I/O Tests

One application, MAESTRO, has been set up to test system I/O capabilities. Two additional runs (512 and 2048 cores) using special input files are required that do the same calculations as in the base case but also write restart files at various points during the run. The objective is to determine how much longer it takes for these two I/O runs than for their non-I/O counterparts. The non-I/O run is to be used for the SSP calculation.

These runs should be done using the "Base Case" run rules. Enter results in **Table 5C**, below.

Full Configuration Test

This test examines the capability and performance of a single application executed over all computational cores and the entire interconnect infrastructure. The supplied test is a custom implementation of a parallel 3D fast Fourier transform using the FFTW (version 2) library for the local part of the transform. The test shall employ the maximum number of MPI tasks feasible on the system and every computational core on the system shall execute at least one MPI task. Therefore, the FFT problem size, which is calculated by the program, will depend on the number of MPI tasks employed and the available system memory. Since the test represents the way that 3-D parallel FFTs are computed in certain electronic structure codes at NERSC it is not permissible to replace the entire test with a library routine; libraries can be used for the FFTW portion only.

ESP

The Effective System Performance (ESP) test measures the performance of resource management and other system management aspects. A description of this test can be found at, http://www.nersc.gov/projects/esp.php. For this procurement, version 2.2 of the ESP test will be used. The test comprises 230 batch jobs with predetermined run times. The objective is to minimize the elapsed time required to process the jobs. As the elapsed time for each job is fixed, the outcome of the test is not dependent on processor speed but depends on aspects such as job launch time and scheduling efficiency. Furthermore, two special jobs require expedited processing so that features such as preemption and reservation have a significant impact. All jobs execute the same simple MPI application that is designed to run in a fixed elapsed time and detect errors. The source distribution includes two perl scripts to generate the batch jobs and run the test. Both scripts will require modifications for the site-specific and platform-specific parameters. Record the elapsed time and ESP efficiency (as outlined in the README) in **Table 6.**

SSP

The SSP is a derived measure of computational capability relevant to achievable scientific work; it shall be used to validate the system and monitor delivered performance throughout the system lifecycle. The SSP is derived from an application performance figure, P_i expressed in units of GFlops per second per processor. Given a system configured with N computational cores, the SSP is the geometric mean of P_i over all applications, multiplied by N. The floating-point operation count used in calculating P_i for each of the seven NERSC-6 component applications has been pre-determined by the University using a hardware performance counter on a single reference system at NERSC and these values may *not* be altered. The floating-point operation counts are *not* measured on the Offeror's system; only the time is. The reference GFLOP counts are to be found in **Tables 7A** and **7B**, below.

As calculated in the manner given above, the SSP represents an "instantaneous" measure of computational capability as of the date the Offeror's application run times were measured. To represent the cumulative computational capability of a system over a specific period of time, the instantaneous SSP is integrated over that time period by multiplying the instantaneous value by the length of time. The result is expressed in units of TFlops (with a factor of 1000 to convert from GFlops/s to TFlops/s).

If the period of time of interest includes either several technology phases available at different times or technology changes (say, due to software improvements), the SSP is determined for each phase and then time-averaged over the entire period.

Six of the seven P_i values in the SSP originate from LARGE-size runs and one originates from the XL-size run.

In all cases, the number of cores used to calculate the SSP for a given application is the number of hardware cores blocked from other use, rather than the number of MPI tasks. This is particularly important to note in the base case when hardware multithreading is used, or a code must be run on additional cores due to memory limitations. It also applies in the optional optimized case when an application is run unpacked on a node or OpenMP is used.

Result Tables

Table 3. System Description

Enter the system details in this table for each system used in benchmarking. Use the System label to refer to system in the following tables.

System	Processor	Clock/MHz	Interconnect	Total Core Count
A				
B				
C				

For each application run, enter the run time variation in the column marked COV. Copy the proposed elapsed times into Table 7, in preparation for the SSP calculation as described below.

Table 4. Lower Level Test Results

NPB 2.3, Class B, Serial (packed)

	System	Benchmarked Rate	Proposed Rate	Units
BT				MOP/s/process
CG				MOP/s/process
FT				MOP/s/process
LU				MOP/s/process
MG				MOP/s/process
SP				MOP/s/process

NPB 2.4 Class D, 256 tasks

	System	Benchmarked Rate	Proposed Rate	Units
BT				MOP/s/process
CG				MOP/s/process
FT				MOP/s/process
LU				MOP/s/process
MG				MOP/s/process
SP				MOP/s/process

NPB UPC Class D, 256 tasks

	System	Benchmarked Rate	Proposed Rate	Units
FT				MOP/s/process

Chombo AMR Elliptic Solver

	System	Benchmarked Time	Proposed Time	Units
256				seconds
1024				seconds
4096				seconds

PSNAP

# of MPI Tasks Used	System	Benchmarked Average Deviation	Proposed Average Deviation	Units
				percent

STREAMS Triad

	System	Benchmarked Rate	Proposed Rate	Units
Single proc. 30%				MB/s
Single proc. 60%				MB/s
Full node				MB/s

Multi-pong

	System	Benchmarked	Proposed	Units
Maximum Inter-Node Latency Single-Core, Farthest-node pair (1)				microseconds
Minimum Inter-Node Latency Single-Core, Nearest-node pair (2)				microseconds
Maximum Intra-Node Latency Single-Core (3)				microseconds
Minimum Intra-Node Latency Single-Core (4)				microseconds
Maximum Inter-Node Latency Fully-packed Nodes, Nearest-node pair (5)				microseconds
Maximum Inter-Node Latency Fully-packed Nodes, Nearest-node pair (6)				microseconds
Maximum Bandwidth Multi-Core Farthest Nodes (7)				MB/s

In the following section just enter the system label to signify that you ran the tests.

	System			
Netperf		***Enter on Separate Tables and CD***		
IOR				
Metabench				

Table 5A. Application Benchmark Results – Base Case (no optimization - target concurrency or necessary higher concurrency)

MEDIUM SIZE						
Application	*System*	*Target Concurrency*	*Concurrency Used*	*Elapsed Time Benchmarked*	*Elapsed Time Proposed*	*COV*
CAM		56				
GAMESS		256				
GTC		512				
IMPACT-T		256				
MAESTRO		512				
MILC		256				
PARATEC		256				

LARGE SIZE						
Application	*System*	*Target Concurrency*	*Concurrency Used*	*Elapsed Time Benchmarked*	*Elapsed Time Proposed*	*COV*
CAM		240				
GAMESS		1024				
GTC		2048				
IMPACT-T		1024				
MAESTRO		2048				
MILC		1024				
PARATEC		1024				

XL SIZE						
Application	*System*	*Target Concurrency*	*Concurrency Used*	*Elapsed Time Benchmarked*	*Elapsed Time Proposed*	*COV*
MILC		8192				

Table 5B. Application Benchmarks – Optional Optimized Case

MEDIUM SIZE						
Application	*System*	*Target Concurrency*	*Concurrency Used*	*Elapsed Time Benchmarked*	*Elapsed Time Proposed*	*COV*
CAM		56				
GAMESS		256				
GTC		512				
IMPACT-T		256				
MAESTRO		512				
MILC		256				
PARATEC		256				

LARGE SIZE						
Application	*System*	*Target Concurrency*	*Concurrency Used*	*Elapsed Time Benchmarked*	*Elapsed Time Proposed*	*COV*
CAM		240				
GAMESS		1024				
GTC		2048				
IMPACT-T		1024				
MAESTRO		2048				
MILC		1024				
PARATEC		1024				

XL SIZE						
Application	*System*	*Target Concurrency*	*Concurrency Used*	*Elapsed Time Benchmarked*	*Elapsed Time Proposed*	*COV*
MILC		8192				

Table 5C. Application I/O Benchmarks

Application	*System*	*Target Concurrency*	*Concurrency Used*	*Elapsed Time Benchmarked*	*Elapsed Time Proposed*	*COV*
MAESTRO		512				
MAESTRO		2048				

Table 6. Other Tests

Full Configuration	*System*	*Benchmarked*	*Proposed*
Memory per Task/MB			
Dimensions (*n1*, *n2*, *n3*)			
# of MPI tasks			
# of Nodes			
Elapsed Time (seconds)			

ESP	*System*	*Benchmarked*	*Proposed*
System Size (cores)			
Elapsed Time (seconds)			
ESP Efficiency			

Table 7A. Application Performance Table For SSP – Base Case

Application	*Elapsed Time Proposed*	*Target Concurrency*	*Concurrency Used*	*GFlop Count*	P_i
CAM		240		57669	
GAMESS		1024		1183900	
GTC		2048		3639479	
IMPACT-T		1024		399414	
MAESTRO		2048		1122394	
MILC (XL)		8192		7337756	
PARATEC		1024		1206376	

Application Performance:
P_i = GFlopCnt/(Elapsed Time*Concurrency)

Table 7B. Application Performance Table For SSP – Optional Optimized Case

Application	*Elapsed Time Proposed*	*Target Concurrency*	*Alternate Concurrency*	*GFlop Count*	P_i	$Y_i=P_iN$
CAM		240		57669		
GAMESS		1024		1183900		
GTC		2048		3639479		
IMPACT-T		1024		399414		
MAESTRO		2048		1122394		
MILC (XL)		8192		7337756		
PARATEC		1024		1206376		

Table 8. SSP Calculation

	System Size (N)	
	A. Base Case	*B. Optional Optimized Case*
	P_i	P_i
CAM		
GAMESS		
GTC		
IMPACT-T		
MAESTRO		
MILC (XL)		
PARATEC		
SSP in TFLOPS = $\frac{N * \sqrt[7]{\prod_i P_i}}{1000}$		
Coefficient of variation (%)		

Enter the results from Tables 7A and 7B. You MUST provide an SSP calculation for column A, the target concurrency or necessary higher concurrency SSP. You may provide an alternative SSP (column B) if different concurrencies provide a better SSP. Be careful to read the notes above regarding System Size (N) for the optimized case.

Appendix: Example Tables

The following example tables contain values obtained on the NERSC system Franklin (Cray XT4). These examples are provided to help you prepare the tables for an RFP response.

System Description

System	Processor	Clock/GHz	Interconnect	Processor Count
A	***Opteron***	***2.6***	***SeaStar2***	***19344***

Application Benchmarks for Target Concurrency Case

		MEDIUM				
Application	System	Target Concurrency	Concurrency Used	Elapsed Time Benchmarked	Elapsed Time Proposed	COV
CAM	***A***	56	***56***		***1609***	
GAMESS	***A***	256	***256***		***4023***	
GTC	***A***	512	***512***		***1476***	
IMPACT-T	***A***	256	***256***		***3011***	
MAESTRO	***A***	512	***512***		***1417***	
MILC	***A***	256	***256***		***859***	
PARATEC	***A***	256	***256***		***1665***	

		LARGE				
Application	System	Target Concurrency	Concurrency Used	Elapsed Time Benchmarked	Elapsed Time Proposed	COV
CAM	***A***	240	***240***		***408***	
GAMESS	***A***	1024	***1024***		***2478***	
GTC	***A***	2048	***2048***		***1493***	
IMPACT-T	***A***	1024	***1024***		***627***	
MAESTRO	***A***	2048	***2048***		***2570***	
MILC	***A***	1024	***1024***		***922***	
PARATEC	***A***	1024	***1024***		***540***	

		XL				
Application	System	Target Concurrency	Concurrency Used	Elapsed Time Benchmarked	Elapsed Time Proposed	COV
MILC	***A***	**8192**	***8192***		***1269***	

Application Performance Calculation for SSP

Application	*Elapsed Time Proposed*	*Target Concurrency*	*Concurrency Used*	*GFlopCnt*	P_i
CAM	***408***	240	***240***	57669	***0.589***
GAMESS	***2478***	1024	***1024***	1183900	***0.467***
GTC	***1493***	2048	***2048***	3639479	***1.190***
IMPACT-T	***627***	1024	***1024***	416200	***0.622***
MAESTRO	***2570***	2048	***2048***	1122394	***0.213***
MILC (XL)	***1269***	8192	***8192***	7337756	***0.706***
PARATEC	***540***	1024	***1024***	1206376	***2.182***

SSP Calculations

	System Size (N)	***19344***
	Base Case: Target Concurrency Or Necessary Higher Concurrency	*Optional Alternate Concurrency, Optimized Code, or Unpacked Nodes*
	P_i	P_i
CAM	***0.589***	
GAMESS	***0.467***	
GTC	***1.190***	
IMPACT-T	***0.622***	
MAESTRO	***0.213***	
MILC (XL)	***0.706***	
PARATEC	***2.182***	
SSP in TFLOPS = $\frac{N * \sqrt[7]{\prod_i P_i}}{1000}$	***13.1***	
Coefficient of variation	***2%***	